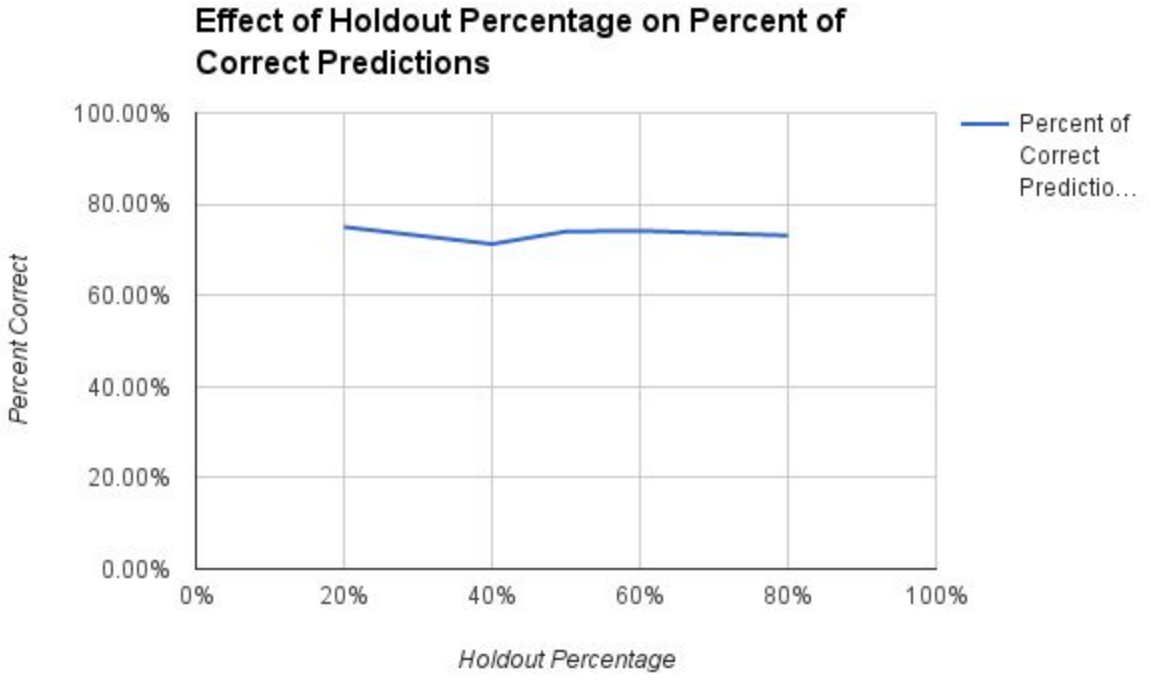
Jacob Hackett and Trevor Rocks
Project 2: Artificial Neural Networks

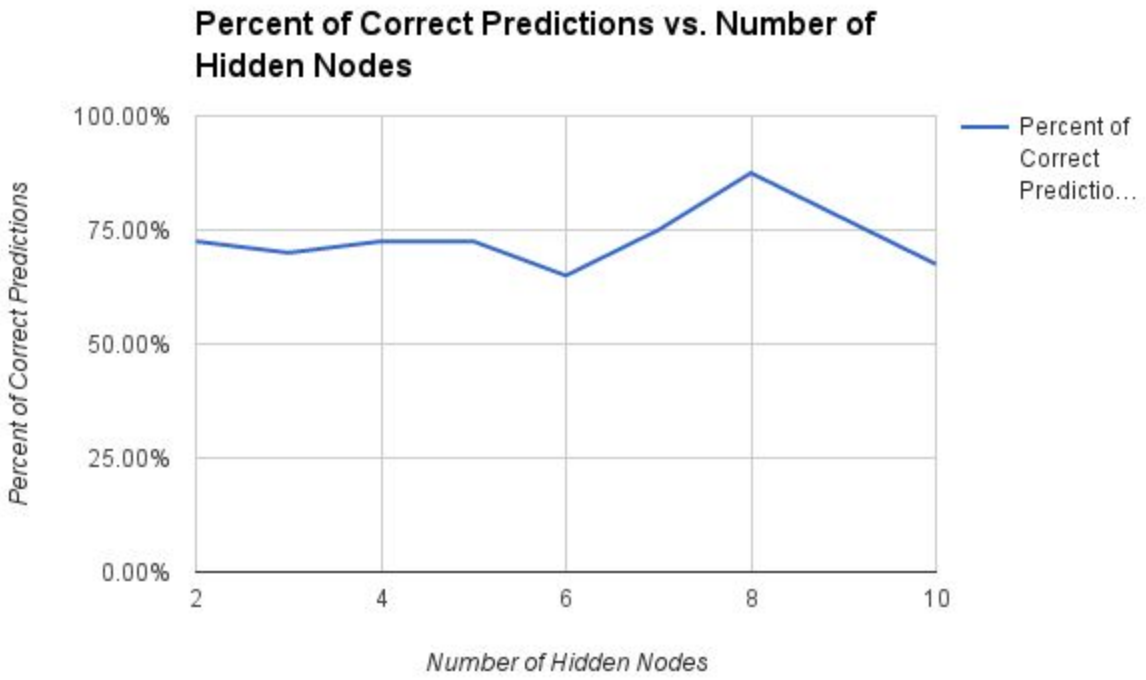
1. Using 5 hidden nodes, we changed the holdout percent and sample. The percent of correct predictions for each run can be seen below. Changing the holdout percent had very little change in the percent of correct predictions.

Holdout Percent	Percent of Correct Predictions
20%	75.000%
40%	71.250%
50%	74.000%
60%	74.167%
80%	73.125%



2. Using 20% holdout percent, we changed the number of hidden nodes and the sample. The percent of correct predictions for each run can be seen below. As you can see, the results range from 65% to 87.5% in no reasonable pattern. It seems as though 8 hidden nodes is the ideal situation in these tests but we would have to run many more tests to be sure.

Number of Hidden Nodes	Percent of Correct Predictions
2	72.500%
3	70.000%
4	72.500%
5	72.500%
6	65.000%
7	75.000%
8	87.500%
9	77.500%
10	67.500%



3. We actually didn't make many simplifying suggestions. Because we implemented the Neural Network as a class, we were able to initialize the system with whatever number of input nodes, output nodes, or hidden nodes that we wanted. I suppose that our system would be relatively reluctant to change its network topography but everything else would be very easy to change (just changing a parameter).

4. Backpropagation is an interesting algorithm. It basically goes through all of the nodes from the output nodes to the hidden nodes to the input nodes and updates the weights of the connections between them all. It does this using error that's calculated using the true value and predicted value for this input set. Over times and after many iterations, the weights of the connections will produce an output much more reliable and consistent with the true value. This happens when the total error is close to zero. However, this takes many, many iterations because simple neural networks are relatively slow learners.
5. The Sigmoid function is used in the neural network to introduce a nonlinear element to the given data, as the sigmoid of a linear data set will return an S shape. The sigmoid is used because it shows the slow starting of the networks learning curve, along with the slow final optimization with the rapid learning rate in the middle. The sigmoid is also preferred because it has a very simple derivative,
   $\frac{d}{dt}Sigmoid(t) = Sigmoid(t)(1 - Sigmoid(t))$.